Data Analysis

Comprehensive Data Cleaning & Exploratory Analysis of Job Market Trends

## Introduction

It is no secret that AI has grown tremendously in availability and impact over the last few years. Students and professionals alike have experienced a rapid indoctrination of AI, whether that be as a simple end-user of ChatGPT or a developer of machine learning models. Being able to understand and adopt these technologies will be indispensable for all industries, and furthermore, open the door to more opportunities with higher compensation.  
  
As Applied Business Analytics students, we understand the importance of not only computer science knowledge, but the application of that knowledge to broader organizational understanding. The following literature review outlines the differences between salaries for AI vs. non-AI careers, regional differences, remote job salary comparisons, and industry comparisons.  
  
This group expects that salaries will be higher for AI careers than for non-AI careers on average. However, unknown is what region and industries have the greatest differences, and whether or not remote status impacts salaries for AI careers.

## Literature Review

In this report, researchers Alekseeva et. al. (Alekseeva et al. ((2021))) recapped their findings of the demand for AI skills in the U.S. economy. The authors point out that it can be difficult to find data regarding companies’ demand for AI skills, so they used job posting data. They defined AI job postings as those that included “artificial intelligence”, “machine vision”, “deep learning” or “speech recognition”. They found that larger firms (measured by market capitalization, cash holdings, and investments in R&D) tended to pay more of a premium for AI skills. Firms with more demand for AI skills also offered high salaries in non-AI jobs than firms with less demand for AI skills.  
  
The results also showed that while IT-related industries have the highest demand for AI skills, there is significant growth across a wide range of industries outside of IT as well. Across all industries, AI skills hold an average of an 11% wage premium. The job with the largest premium is Management; the authors surmise that this means AI has the greatest value when combined with organizational knowledge.  
  
Authors Pabilonia and Vernon outline differences in compensation for remote and in-office roles (Pabilonia and Vernon ((2025))). Using data from American Time Use Survey (ATUS) they note that remote work has increased in the US since 2020 (causal connection with the COVID-19 pandemic), and importantly “remote workers in most occupations earned a wage premium”. The authors performed extensive mathematical analysis within the paper, determining that in 2021 there was a 13.3% premium for remote work versus in-office work. Wages within the paper are indicated as being “determined by a number of factors, including job tasks, productivity differences, compensating differentials for job amenities, search frictions, and monopsony power, among others”. The authors also touched on disparity in pay within the principal city of the large metropolitan statistical area (MSA), noting that the wage premium for individuals working within the MSA was smaller (12.6%) than those working outside the 15 largest MDAs (14.5%).  
  
Card et. al. (Card, Rothstein, and Yi ((2023))) discuss the locale for the highest paid jobs within the US. Unsurprisingly, using “data from the Longitudinal Employer-Household Dynamics program” the authors demonstrate connections between location and salary. Regions in major metropolitans and large industries have wage premiums up to 18% higher than national averages. The authors theorize that larger cities pay more due to their ability to attract and retain higher-skilled workers and the likely presence of large companies to support the talent. The authors do however also find that despite the higher wages in the major commuting zones (CZs), local costs “fully offset local pay premiums, implying that workers who move to larger CZs have no higher net-of-housing consumption”. This leads to the understanding that while more highly compensated in major metropolitan areas, the cost of living consumes the wage premium.  
  
According to the U.S. Bureau of Labor Statistics (BLS) (Labor Statistics ((2024))), total employment is projected to grow by 4 percent between 2023 and 2033. There is also an expected increase in the number of jobs from 2023 to 2033 from 167.8 million to 17.46 million. Half of the forecasted job gains are expected to be in sectors such as healthcare, scientific and technical services. Whereas, the retail sector is more likely to lose jobs over the coming years.  
  
Industries that will see the most significant wage growth from 2023 and 2033 are those that will have a strong demand for specialized labor. Healthcare and social assistance jobs will see a rise in high paying roles such as nurse practitioners and physical therapists assistants. This sector will experience continued growth due to skill labor shortages and aging population. The scientific and technical services sector is expected to grow by 10.5% and it includes high paying roles such as software developers, cybersecurity professionals and data scientists. Renewable Energy industries sector will see an increase in jobs over the years as demand for photovoltaic installers and wind turbines service technicians is increasing and are expected to be hired at competitive rates.

## References

# Exploratory Data Analysis

## Importing Dataset Using Pandas

import pandas as pd  
  
# Load the dataset  
data = pd.read\_csv('./data/lightcast\_job\_postings.csv')

## Dropping Unncessary Columns

* **Which columns are irrelevant or redundant?**  
  ID, URL, ACTIVE\_URLS, DUPLICATES, LAST\_UPDATED TIMESTAMP are irrelevant or redundant columns. They are mostly used for internal tracking and don’t contribute to the actual analysis of jobs, industries, and occupations.
* **Why are we removing multiple versions of NAICS/SOC codes?**  
  The dataset contains multiple versions of industry NAICS and Occupational SOC which can be risky to keep, as there is risk of duplications and inconsistent groupings.
* **How will this improve analysis?**  
  This will improve the analysis by enhancing the efficiency of the data; by having smaller datasets it will be easier to process and run data. It will also improve consistency because by having only one version of the data the risk of duplication will be low.

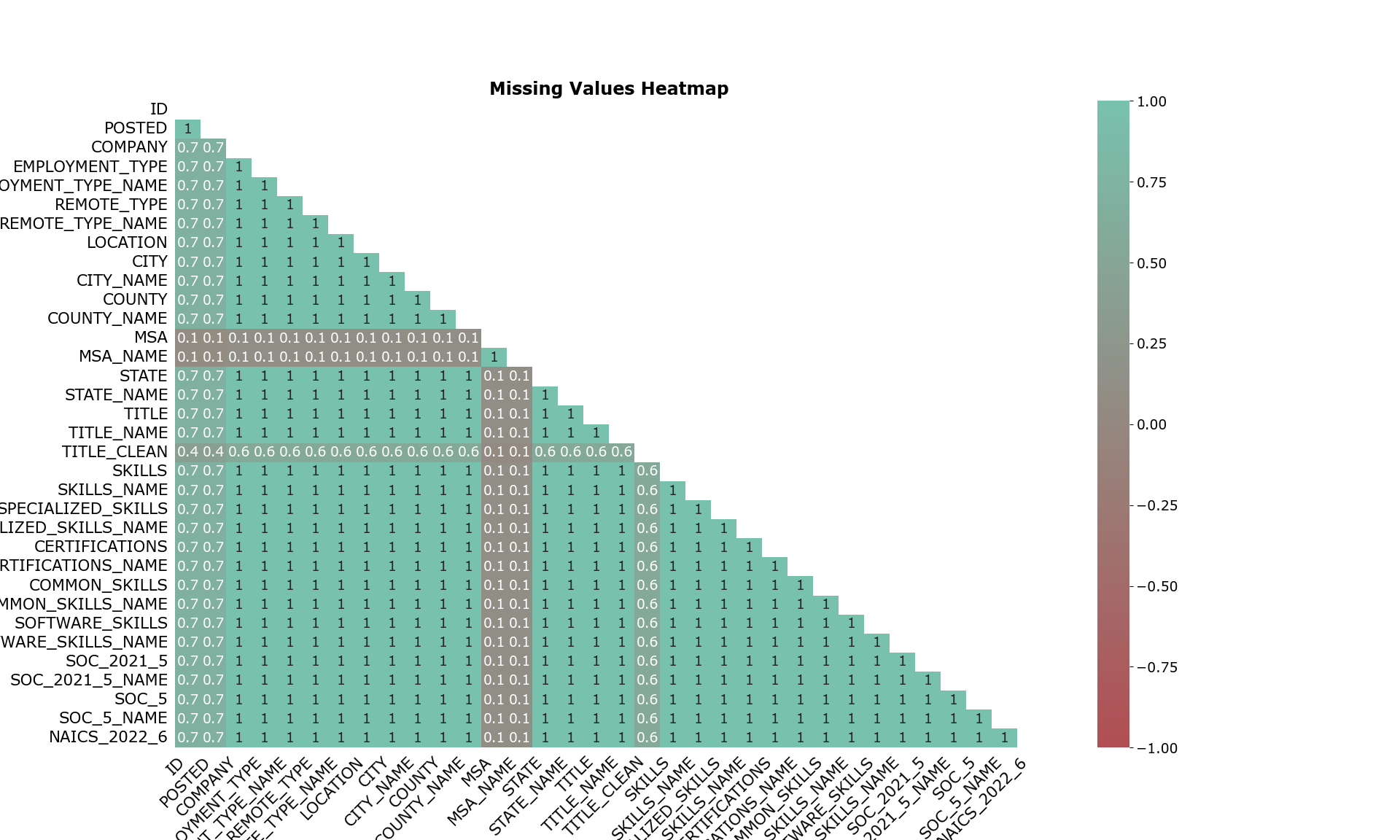
columns\_to\_drop = [  
'LAST\_UPDATED\_DATE', 'LAST\_UPDATED\_TIMESTAMP', 'DUPLICATES', 'EXPIRED', 'ACTIVE\_SOURCES\_INFO', 'TITLE\_RAW','BODY',  
'COMPANY\_NAME', 'COMPANY\_RAW', 'COMPANY\_IS\_STAFFING', 'EDUCATION\_LEVELS', 'EDUCATION\_LEVELS\_NAME', 'MIN\_EDULEVELS',  
'MIN\_EDULEVELS\_NAME', 'MAX\_EDULEVELS', 'MAX\_EDULEVELS\_NAME', 'MIN\_YEARS\_EXPERIENCE', 'MAX\_YEARS\_EXPERIENCE', 'IS\_INTERNSHIP',  
'ORIGINAL\_PAY\_PERIOD', 'SALARY\_TO', 'SALARY\_FROM', 'COUNTY\_OUTGOING', 'COUNTY\_NAME\_OUTGOING', 'COUNTY\_INCOMING', 'COUNTY\_NAME\_INCOMING',  
'MSA\_OUTGOING', 'MSA\_NAME\_OUTGOING','MSA\_INCOMING', 'MSA\_NAME\_INCOMING', 'NAICS2', 'NAICS2\_NAME', 'NAICS3', 'NAICS3\_NAME', 'NAICS4','NAICS4\_NAME',  
'NAICS5', 'NAICS5\_NAME', 'NAICS6', 'NAICS6\_NAME', 'ONET', 'ONET\_NAME', 'ONET\_2019', 'ONET\_2019\_NAME', 'CIP6', 'CIP6\_NAME', 'CIP4', 'CIP4\_NAME',  
'CIP2', 'CIP2\_NAME', 'SOC\_2021\_2', 'SOC\_2021\_2\_NAME', 'SOC\_2021\_3', 'SOC\_2021\_3\_NAME', 'SOC\_2021\_4', 'SOC\_2021\_4\_NAME', 'LOT\_OCCUPATION\_GROUP\_NAME',  
'LOT\_V6\_SPECIALIZED\_OCCUPATION', 'LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION', 'LOT\_V6\_OCCUPATION\_NAME', 'LOT\_V6\_OCCUPATION\_GROUP',  
'LOT\_V6\_OCCUPATION\_GROUP\_NAME', 'LOT\_V6\_CAREER\_AREA', 'LOT\_V6\_CAREER\_AREA\_NAME', 'SOC\_2', 'SOC\_2\_NAME', 'SOC\_3', 'SOC\_3\_NAME', 'SOC\_4', 'SOC\_4\_NAME',  
'NAICS\_2022\_2', 'NAICS\_2022\_2\_NAME', 'NAICS\_2022\_3', 'NAICS\_2022\_3\_NAME', 'NAICS\_2022\_4', 'NAICS\_2022\_4\_NAME', 'NAICS\_2022\_5', 'NAICS\_2022\_5\_NAME',  
'DURATION', 'SOURCE\_TYPES', 'SOURCES', 'URL', 'ACTIVE\_URLS', 'MODELED\_EXPIRED', 'MODELED\_DURATION', 'LOT\_CAREER\_AREA', 'LOT\_CAREER\_AREA\_NAME',  
'LOT\_OCCUPATION', 'LOT\_OCCUPATION\_NAME', 'LOT\_SPECIALIZED\_OCCUPATION', 'LOT\_SPECIALIZED\_OCCUPATION\_NAME', 'LOT\_OCCUPATION\_GROUP', 'LIGHTCAST\_SECTORS',  
'LIGHTCAST\_SECTORS\_NAME'  
]  
data.drop(columns=columns\_to\_drop, inplace=True)

## Handling Missing Values

* **Answer the question: How should missing values be handled?**
  + Numerical fields (e.g., Salary) are filled with the median.
  + Categorical fields (e.g., Industry) are replaced with “Unknown”.
  + Columns with >50% missing values are dropped.

import missingno as msno  
import matplotlib.pyplot as plt  
  
  
  
# Fill missing values  
data["SALARY"].fillna(data["SALARY"].median(), inplace=True)  
data["NAICS\_2022\_6\_NAME"].fillna("Unknown", inplace=True)  
data["REMOTE\_TYPE\_NAME"].fillna("None, inplace=True")  
  
data.rename(columns={'NAICS\_2022\_6\_NAME': 'INDUSTRY'}, inplace=True)

data.dropna(thresh=len(data) \* 0.5, axis=1, inplace=True)  
  
# Visualize missing data  
import matplotlib as mpl  
from matplotlib.colors import LinearSegmentedColormap  
  
# Set global font settings  
mpl.rcParams['font.family'] = 'Verdana'  
mpl.rcParams['font.size'] = 14  
mpl.rcParams['text.color'] = 'black'  
mpl.rcParams['axes.labelcolor'] = 'black'  
mpl.rcParams['xtick.color'] = 'black'  
mpl.rcParams['ytick.color'] = 'black'  
  
# Define custom green-to-red colormap  
custom\_cmap = LinearSegmentedColormap.from\_list(  
 'custom\_green\_red', ['#B14E53', '#78C2AD'], N=256  
)  
  
# Create the heatmap  
fig = plt.figure(figsize=(10, 6))  
ax = msno.heatmap(data, cmap=custom\_cmap)  
  
# Set the title  
plt.title("Missing Values Heatmap", fontsize=18, fontweight='bold')  
  
plt.savefig("figures/missing\_values.png")  
plt.show()



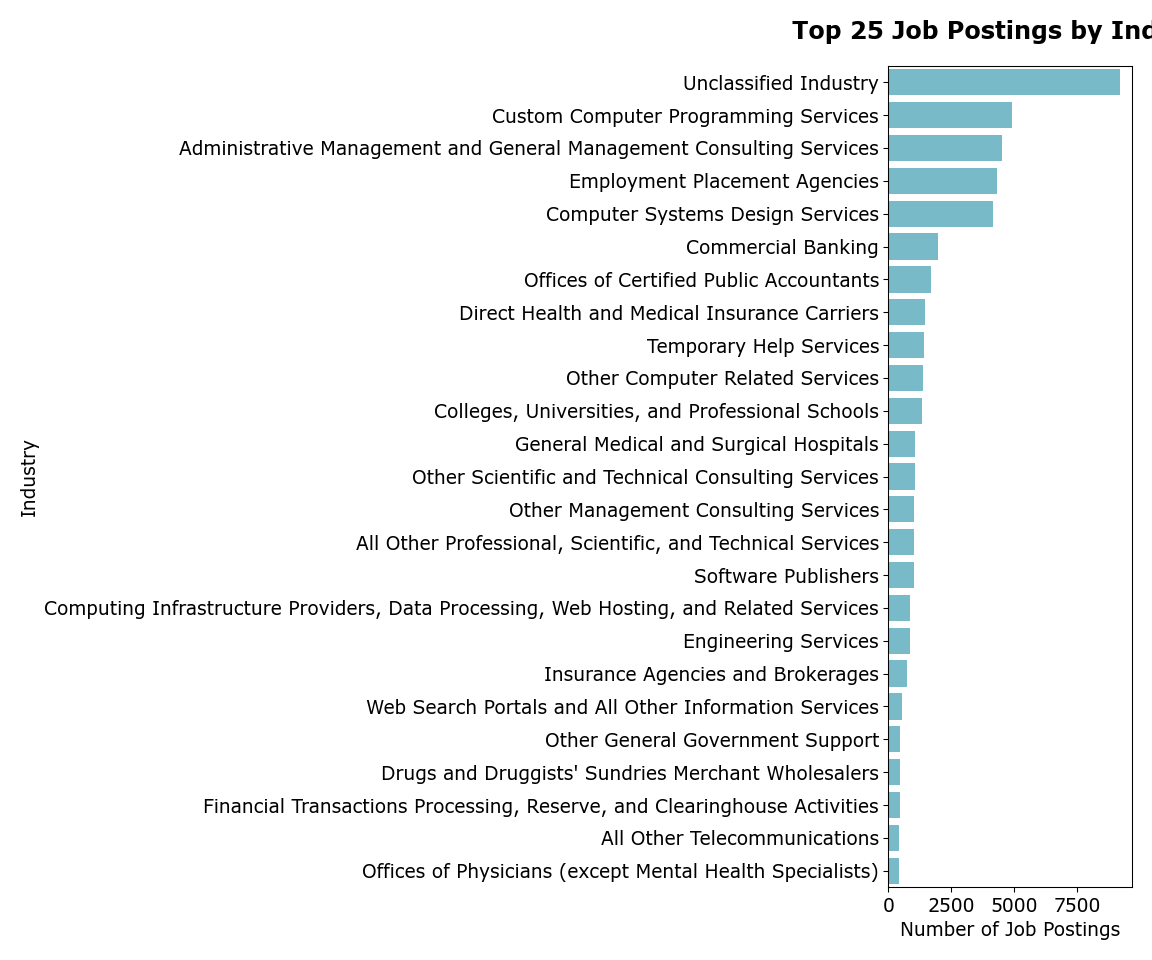
The heatmap shows a few missing values in the dataset. In this dataset most fields cluster near 1. The value 1 (dark blue) means the two columns have missing values together. 0.0 value (white) suggest there is no relationship.

## Remove Duplicates

data = data.drop\_duplicates(subset=["TITLE", "COMPANY", "LOCATION", "POSTED"], keep="first")

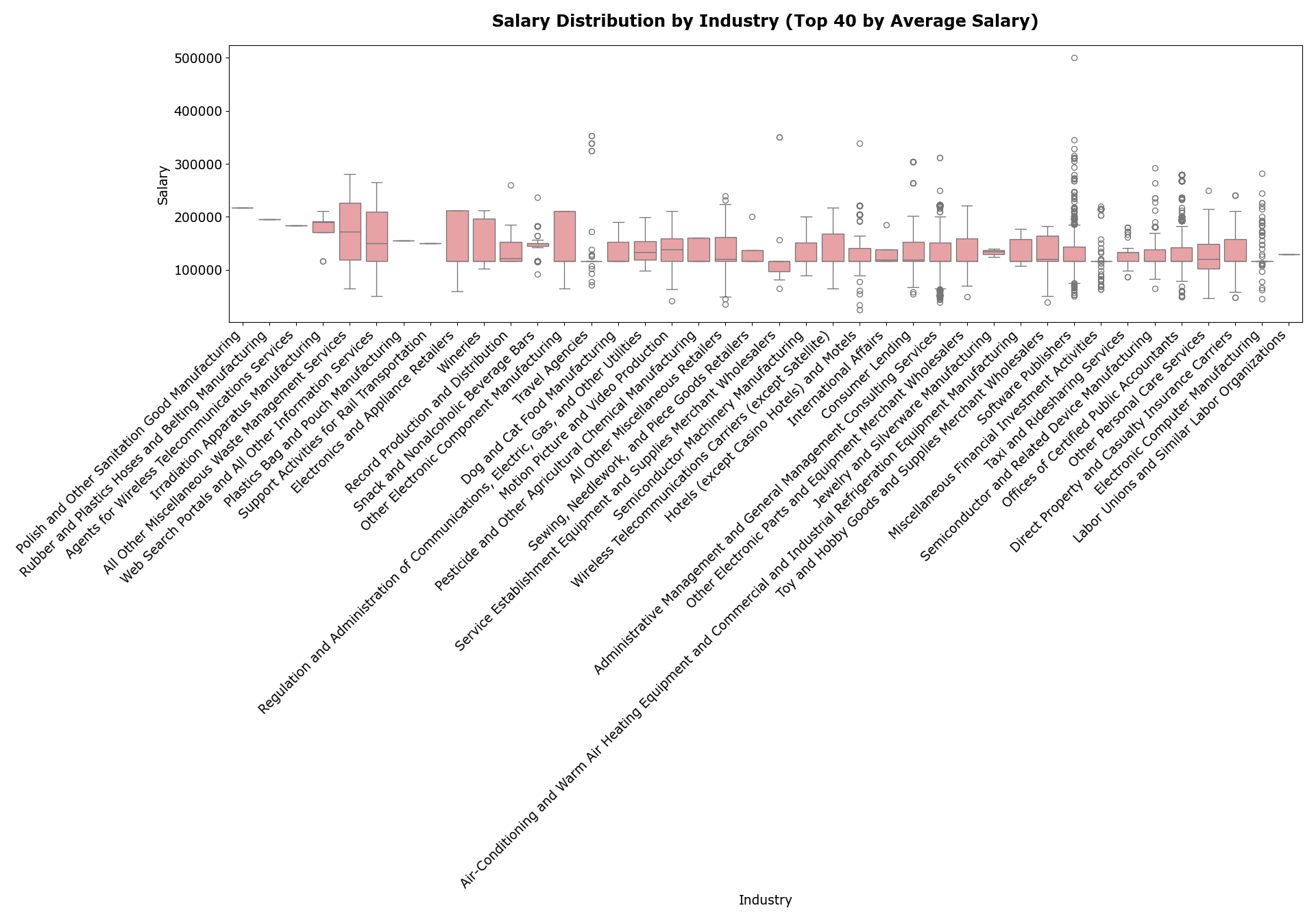
## Job Postings by Industry

import seaborn as sns  
import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
# Set global font family to Verdana  
mpl.rcParams['font.family'] = 'Verdana'  
  
# Compute posting count per industry  
data["posting\_count"] = data["ID"].groupby(data["INDUSTRY"]).transform("count")  
  
# Summarize and sort top 25 industries  
industry\_summary = data.groupby("INDUSTRY")["posting\_count"].first().sort\_values(ascending=False).head(25)  
  
# Plot  
plt.figure(figsize=(12, 10))  
sns.barplot(  
 x=industry\_summary.values,  
 y=industry\_summary.index,  
 orient='h',  
 color='#6CC3D5'  
)  
  
# Title and labels with updated font sizes  
plt.title("Top 25 Job Postings by Industry", fontsize=18, fontweight='bold', pad=20)  
plt.xlabel("Number of Job Postings", fontsize=14)  
plt.ylabel("Industry", fontsize=14)  
  
plt.tight\_layout()  
  
plt.savefig("figures/top\_postings.png")  
plt.show()



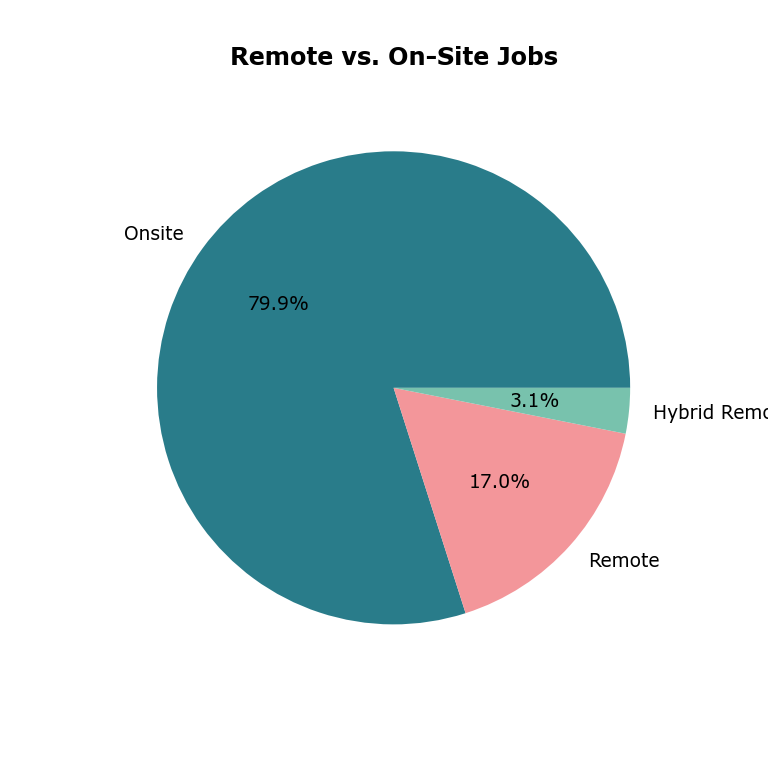
## Salary Distribution by Industry

import seaborn as sns  
import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
# Set global font family to Verdana  
mpl.rcParams['font.family'] = 'Verdana'  
  
# Calculate average industry salary  
data["AVERAGE\_INDUSTRY\_SALARY"] = data["SALARY"].groupby(data["INDUSTRY"]).transform("mean").round()  
  
# Get top 40 industries by average salary  
top\_40\_industries = data.groupby("INDUSTRY")["AVERAGE\_INDUSTRY\_SALARY"].first().sort\_values(ascending=False).head(40).index  
filtered\_data = data[data["INDUSTRY"].isin(top\_40\_industries)]  
  
# Create figure with taller height  
plt.figure(figsize=(20, 14)) # Increased height from 10 to 14  
  
# Create boxplot with custom color  
sns.boxplot(  
 data=filtered\_data,  
 x="INDUSTRY",  
 y="SALARY",  
 order=top\_40\_industries,  
 color='#F3969A' # Custom box color  
)  
  
# Title and labels with updated font sizes  
plt.title("Salary Distribution by Industry (Top 40 by Average Salary)", fontsize=18, fontweight='bold', pad=20)  
plt.xlabel("Industry", fontsize=14)  
plt.ylabel("Salary", fontsize=14)  
  
# Rotate x-tick labels for readability  
plt.xticks(rotation=45, ha='right')  
  
plt.tight\_layout()  
  
  
plt.savefig("figures/salary\_dist.png")  
plt.show()



## Remote vs. On-Site Jobs

import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
# Set global font family to Verdana  
mpl.rcParams['font.family'] = 'Verdana'  
  
# Clean and group remote types  
remote\_grouped = data["REMOTE\_TYPE\_NAME"].fillna("Onsite").replace({  
 "[None]": "Onsite",   
 "Not Remote": "Onsite",  
 "None": "Onsite"  
})  
  
# Count values  
remote\_counts = remote\_grouped.value\_counts()  
  
# Define colors matching labels (make sure the order matches remote\_counts.index)  
color\_map = {  
 "Onsite": "#297C8A",  
 "Hybrid Remote": "#78C2AD",  
 "Remote": "#F3969A"  
}  
  
# Get colors for the pie chart slices in the correct order, fallback to gray if missing  
colors = [color\_map.get(label, "#cccccc") for label in remote\_counts.index]  
  
# Plot pie chart  
plt.figure(figsize=(8, 8))  
wedges, texts, autotexts = plt.pie(  
 remote\_counts.values,  
 labels=remote\_counts.index,  
 autopct='%1.1f%%',  
 colors=colors,  
 textprops={'fontsize': 14, 'color': 'black'}  
)  
  
# Title with larger font  
plt.title("Remote vs. On-Site Jobs", fontsize=18, fontweight='bold', pad=20)  
  
# Adjust autotext (percentage text) font size  
for autotext in autotexts:  
 autotext.set\_fontsize(14)  
  
  
plt.savefig("figures/remote\_onsite.png")  
plt.show()



## Why these visualizations were chosen

* **Bar Chart: Job Postings by Industry**
  + Makes it easy to compare job postings across different industries.
  + Provides a clear ranking that is simple to interpret.
* **Boxplot: Salary Distribution by Industry**
  + Shows medians, outliers, and summarizes salary distributions.
  + Allows for comparisons between industries and helps detect salary variability.
* **Pie Chart: Job Location Types**
  + Provides a clear visual breakdown of remote versus on-site jobs within the dataset.

## Key insights from each graph

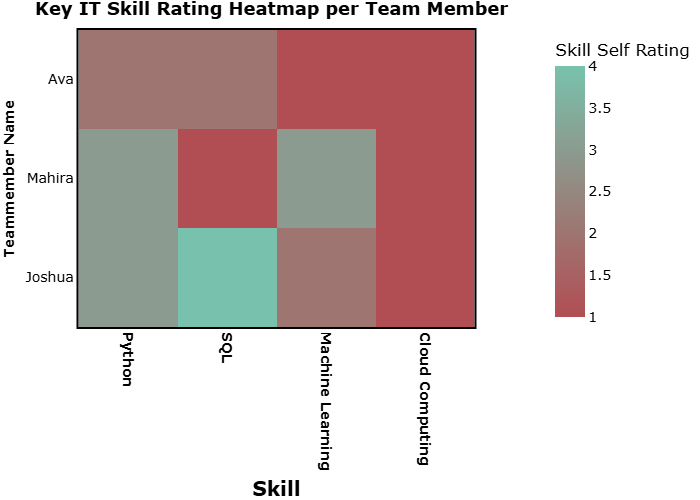
* **Bar Chart: Top 25 Industries by Job-Posting Volume**
  + The distribution is skewed, showing demand concentrated in technology and professional services.
  + A large portion of “unclassified industry” job postings suggests many jobs are not mapped to any industry.
* **Boxplot: Salary Distribution by Industry**
  + Salaries across the 40 industries skew high, with medians generally in the $120k–$170k range.
  + Industries with higher medians and greater dispersions include software/semiconductors and consulting.
  + Several industries have wide IQRs and many upper outliers, while some have less variability, indicating standardized pay.
* **Pie Chart: Job Location Types**
  + 17% of jobs are remote, 3.1% are hybrid, and 1.6% are not remote.
  + A large number of job postings do not specify whether the job is remote or on-site.

## Team Skill Level Dataframe

import pandas as pd  
  
skills\_data = {  
 "Name": ["Ava", "Mahira", "Joshua"],  
 "Python": [2, 3, 3],  
 "SQL": [2, 1, 4],  
 "Machine Learning": [1, 3, 2],  
 "Cloud Computing": [1, 1, 1]  
}  
  
df\_skills = pd.DataFrame(skills\_data)  
df\_skills.set\_index("Name", inplace=True)  
df\_skills

## Visualizing Skill Gaps in Heatmap with Plotly

import kaleido  
import plotly.express as px  
custom\_colorscale = ["#B14E53", "#78C2AD", ]  
fig = px.imshow(df\_skills,  
 labels=dict(x="Skill", y="Team Member Name", color="Skill Self Rating"),  
 title="Key IT Skill Rating Heatmap per Team Member",  
 color\_continuous\_scale=custom\_colorscale  
)  
fig.update\_layout(  
 title=dict(  
 text="Key IT Skill Rating Heatmap per Team Member",  
 font=dict(size=18, family="Verdana", color="black", weight="bold")  
 ),  
 xaxis=dict(  
 title=dict(text="Skill", font=dict(size=20, family="Verdana", color="black", weight="bold")),  
 tickangle=90,  
 tickfont=dict(size=14, family="Verdana", color="black", weight="bold"),  
 showline=True,  
 linewidth=2,  
 linecolor="black",  
 mirror=True,  
 showgrid=True,  
 ),  
 yaxis=dict(  
 title=dict(text="Teammember Name", font=dict(size=14, family="Verdana", color="black", weight="bold")),  
 showline=True,  
 linewidth=2,  
 linecolor="black",  
 mirror=True,  
 showgrid=True,  
 gridcolor="white",  
 gridwidth=0.5  
 ),  
 font=dict(family="Verdana", size=14, color="black"),  
 paper\_bgcolor="white",  
 showlegend=True,  
)  
  
fig.write\_image("figures/skill\_heatmap.png")  
fig.write\_html("figures/skill\_heatmap.html")  
fig.show()

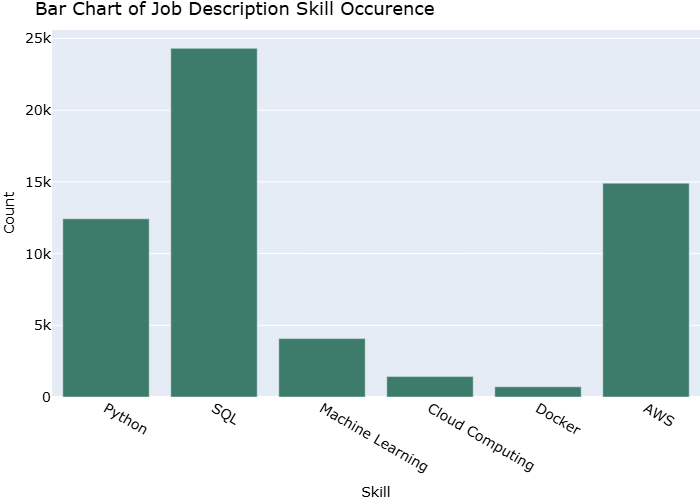


title: “Skill Gap Analysis” author: - name: Mahira Ayub affiliations: - id: bu name: Boston University city: Boston state: MA - name: Ava Godsy affiliations: - ref: bu - name: Joshua Lawrence affiliations: - ref: bu execute: eval: false —

import pandas as pd  
df = pd.read\_csv("data/lightcast\_job\_postings.csv")  
  
filtered\_df = df[["ID", "BODY"]]  
filtered\_df = filtered\_df.dropna()

# Assuming job\_descriptions is a list of text from job postings  
top\_skills = ["Python", "SQL", "Machine Learning", "Cloud Computing", "Docker", "AWS"]  
  
skill\_counts = {}  
for skill in top\_skills:  
 skill\_counts[skill] = filtered\_df["BODY"].str.contains(skill, case=False, na=False).sum()  
  
skills\_df\_pd = pd.DataFrame(list(skill\_counts.items()), columns=['Skill', 'Count'])  
  
print(skills\_df\_pd)  
skills\_df\_pd.to\_csv("data/skills\_job\_description.csv", index=False)

import plotly.express as px  
  
top\_skills\_fig = px.bar(  
 skills\_df\_pd,  
 x='Skill',  
 y='Count',  
 title='Bar Chart of Job Description Skill Occurence',  
 color\_discrete\_sequence=['#3D7C6A'] # Set bar color  
)  
  
# Update layout for fonts and sizes  
top\_skills\_fig.update\_layout(  
 font=dict(  
 family="Verdana",  
 size=14,  
 color="black"  
 ),  
 title=dict(  
 font=dict(  
 family="Verdana",  
 size=18,  
 color="black"  
 )  
 ),  
 xaxis=dict(  
 title\_font=dict(  
 family="Verdana",  
 size=14,  
 color="black"  
 )  
 ),  
 yaxis=dict(  
 title\_font=dict(  
 family="Verdana",  
 size=14,  
 color="black"  
 )  
 )  
)  
  
top\_skills\_fig.show()  
top\_skills\_fig.write\_image("./figures/top\_skills.png")  
top\_skills\_fig.write\_html("./figures/top\_skills.html")



# Which skills should each member prioritize learning?

* Based on the self assessment rating of each member of the group, everyone should prioritize learning cloud computing.
* Cloud Computing is gaining popularity as data is moving to the cloud. Most companies are using cloud services to have easy access to data globally and in real time.

# What courses or resources can help?

* The resources and courses that can be helpful in learning cloud computing skills include:
  + Google Cloud Fundamentals
  + AWS Academy Cloud Foundations
  + Microsoft Learn - Azure Fundamentals
  + AWS Certified Data Analytics – Specialty
  + Azure Data Fundamentals (DP-900)

# How can the team collaborate to bridge skill gaps?

* Identify gaps that are related to the project needs and work on it as a group to learn from eachother.
* Split tasks and divide work according to each memeber’s skill level.
* Cross-train so everyone can learn from each other and develop understanding of cloud analytics tasks.

Alekseeva, L., J. Azar, M. Giné, S. Samila, and B. Taska. (2021): “[The demand for AI skills in the labor market](https://doi.org/10.1016/j.labeco.2021.102002),” *Labour Economics*, 71, 102002.

Card, D., J. Rothstein, and M. Yi. (2023): [Location, Location, Location](https://doi.org/10.3386/w31587), Working Paper,. Working paper seriesNational Bureau of Economic Research.

Labor Statistics, B. of. (2024): “Industry and occupational employment projections overview and highlights, 2023–33,” *Monthly Labor Review*,.

Pabilonia, S. W., and V. Vernon. (2025): “[Remote work, wages, and hours worked in the united states](https://doi.org/10.1007/s00148-025-01064-9),” *Journal of Population Economics*, 38.